

Sparse Phase Retrieval via Reweighted Thresholded Wirtinger Flow

Lan Li, Yufeng Ren, Kai Yu

Abstract—In engineering, most sensors can only measure the intensity information of a signal, but without the phase information. The phase retrieval problem is to reconstruct the original signal only from intensity values, which is an ill-posed problem. To address this issue, we introduce sparse prior information to overcome the ill-posed problem and propose the sparse reweighted thresholded Wirtinger flow (SRThWF) algorithm. In the SRThWF algorithm, the original signal is reconstructed from Gaussian random measurements via intensity values. In particular, SRThWF carries on iteratively in two stages: In stage one, it works with initialization by truncated spectral method restricted on support recovery of sparse signal. In the second stage, the initial estimate is refined iteratively via hard-thresholding-based adaptive reweighted gradient descent. Furthermore, the convergence of the proposed algorithm has been proved. Experimental results show that the proposed algorithm is an effective sparse phase retrieval solver in the probability of success even if the sparsity level is unknown. Not only that, the proposed algorithm exhibits a faster convergence rate and shorter running time. In addition, SRThWF is robust to the noise.

Index Terms—sparse phase retrieval, convergence, gradient descent, thresholded, reweighted

I. INTRODUCTION

A. Phase retrieval problem

PHASE retrieval (PR) is an inverse problem that reconstructs the original signal only from the intensity measurement values. This problem arises in various scientific and engineering fields, including microscopy [1], X-ray crystallography [2], and astronomy [3]. The mathematical model for the phase retrieval problem can be expressed as:

$$y_i = |\langle \mathbf{a}_i, \mathbf{x} \rangle|^2 + \varphi_i, \quad i = 1, \dots, m, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the original signal, $\{\mathbf{a}_i\}_{i=1}^m \in \mathbb{R}^n$ are the measurement vectors, $\{y_i\}_{i=1}^m \in \mathbb{R}^n$ are the intensity values, $\{\varphi_i\}_{i=1}^m$ are the noise, and m corresponds to the number of measurements, also referred to as the sample size. In other words, the goal of the PR problem is to reconstruct

\mathbf{x} from $\{y_i\}_{i=1}^m$.

In general, the PR problem involves reconstructing the original signal from either Fourier measurements or Gaussian random measurements. For instance, Fourier measurements can be attributed to the [4] and [5]. The Gerchberg-Saxton (GS) [4] algorithm is the first solution developed to solve the PR problem. Lan Li [5] demonstrates the connectivity of directed graphs and provides the necessary and sufficient conditions for recovering the phase from multiple short-time Fourier measurements. Furthermore, the PR problem is discussed in the context of random measurements. In the last few years, more and more optimization methods have emerged for addressing the PR problem from Gaussian random measurements. Among these methods, the Wirtinger flow (WF) [6] algorithm stands out as an effective solution. It employs Wirtinger derivatives in gradient descent to iteratively refine the initial estimate. Following the spirit of WF algorithm, the truncated Wirtinger flow (TWF) [7] algorithm adopts an adaptive gradient flow-based truncation rule. Unlike the TWF method, the reweighted Wirtinger flow (RWF) [8] algorithm introduces an indirect adaptive truncation strategy by adding weights. Nevertheless, it is a challenging problem to reconstruct the original signal \mathbf{x} only from intensity values $\{y_i\}_{i=1}^m$ without any prior information about \mathbf{x} .

B. Sparse prior information

With the discussion of the previous section, it is essential to consider the role of prior information, such as sparsity [9] and neural network priors [10], in addressing the phase retrieval problem. Owing to the practical significance of sparse prior information, it is necessary to solve the sparse PR problem. Moreover, it has been demonstrated that $m \geq 4k - 1$ measurements are necessary and sufficient to ensure the uniqueness of the PR problem for k sparse real signals with a high probability [11]. A series of heuristic methods have been introduced to address the PR problem for sparse signals.

Sparse AltMinPhase [12] algorithm alternately updates between phase information and the candidate solution with resampling. Inspired by the generalization of WF, the authors present thresholded Wirtinger flow (ThWF), sparse Wirtinger flow (SWF), and sparse truncated amplitude flow (SPARTA) algorithms in [13] [14] [15], respectively. In particular, SPARTA has a higher probability of success and a faster convergence rate than ThWF and sparse AltMinPhase. However, the performance of SPARTA is significantly impacted by the sparse prior information, and the truncated process might fail to consider some valuable information. In contrast, SWF demonstrates greater robustness to unknown

Manuscript received August 5, 2023; revised February 19, 2024.

This work is supported by the Natural Science Basic Program of Shannxi (Program No.2021JM-399), and the Graduate Innovation and Practical Ability Training Project of Xi'an Shiyou University (YCS22213173).

Lan Li is a professor at the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: lanli@xsyu.edu.cn).

Yufeng Ren is a postgraduate student of the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: yf09876188@163.com).

Kai Yu is a postgraduate student of the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: jamesyu0228@gmail.com).

sparse levels compared to SPARTA.

In this work, a sparse reweighted thresholded Wirtinger flow (SRThWF) algorithm is proposed to solve the sparse phase retrieval problem. The initialization stage uses an analytically well-justified rule to estimate the support of the sparse signal. Then, the initialization is obtained via the truncated spectral method that is approximate to the original sparse signal. Eventually, weights are added to the gradient descent based on hard thresholding. Our innovations in this work are as follows:

A sparse reweighted thresholded Wirtinger flow (SRThWF) algorithm merges the ideas from adding weights and the SWF algorithm. Adaptive weights are induced during the gradient descent process to accelerate the convergence rate. Specifically, when the weight $\{w_i\}_{i=1}^m$ values are equal to 1, this is a generalization of SWF [14]. On the theoretical side, this paper offers a convergence proof for the proposed algorithm.

Concerning the standard notation consistently employed in this paper, the bold uppercase and lowercase letters indicate matrices and vectors, respectively. $\|\bullet\|_0$ represents the number of non-zero elements. T illustrates transpose. The inner product of vector \mathbf{a} and vector \mathbf{b} is defined as $\langle \mathbf{a}, \mathbf{b} \rangle$. $|\bullet|$ denotes absolute value. $\lceil \bullet \rceil$ returns the smallest integer greater than or equal to the given number. Let S be the support set for a k -sparse signal. $B \setminus P$ shows the differences between two sets B and P .

II. THE PROPOSED ALGORITHM AND CONVERGENCE PROOF

In this section, the sparse PR problem is tackled with the introduction of the SRThWF algorithm. There are mainly two stages: In stage one, the support estimation of sparse signal needs to be performed from the law of large numbers. In addition, a truncated spectrum method for initialization is obtained via measurement vectors restricted on the support; and in stage two, the approach of reweighting is employed for the gradient descent process to update the initialization. Next, we will provide a detailed elaboration on these two stages.

A. SRThWF Algorithm

The mathematical model for the sparse PR problem is as follows:

$$\begin{aligned} & \text{Find } \mathbf{z} \\ & y_i = |\langle \mathbf{a}_i, \mathbf{z} \rangle|^2, \quad i = 1, 2, \dots, m, \\ & \|\mathbf{z}\|_0 = k, \end{aligned} \quad (2)$$

where \mathbf{z} represents an approximation of the original signal \mathbf{x} , and k represents the number of non-zero elements of the signal \mathbf{z} . Nevertheless, the l_0 -norm problem is NP-hard. Consequently, the sparse reweighted thresholded Wirtinger flow model can be expressed:

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^n} f(\mathbf{z}) &= \frac{1}{2m} \sum_{i=1}^m w_i (|\mathbf{a}_i^T \mathbf{z}|^2 - y_i)^2, \quad w_i \geq 0, \\ \text{s.t. } \|\mathbf{z}\|_0 &= k \end{aligned} \quad (3)$$

where $\{w_i\}_{i=1}^m$ are weight parameters.

Stage1: initialization restricted on support estimation

The number of elements in the support set S is indicated

by its cardinality $|S|$ as $|S| = k \ll n$. The j -th element of vector \mathbf{a}_i is defined as $a_{i,j}$, and we have the following formula:

$$Q_{i,j} = y_i^2 a_{i,j}^2, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (4)$$

Note that the vectors \mathbf{a}_i i.i.d. $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. According to the rotational invariance of the Gaussian distribution, $Q_{i,j}$ have the following equation:

$$\mathbb{E}(Q_{i,j}) = 2x_j^2 + \|\mathbf{x}\|_2^2, \quad (5)$$

where $\mathbb{E}(\bullet)$ expresses expectation. From the formula (5), for all j , the support set S can be obtained when $2x_j^2$ is sufficiently large. Then, \hat{S} is fixed as the set of corresponding indices of k largest values in $\hat{Q}_{i,j} = \frac{1}{m} \sum_{i=1}^m Q_{i,j}$. As m increases, $\hat{Q}_{i,j}$ is approximately equivalent to $\mathbb{E}(Q_{i,j})$.

Therefore, we can estimate the support using the formula:

$$\hat{S} = \{1 \leq j \leq n \mid \text{index of top-}k \text{ instances in } \{\hat{Q}_{i,j}\}_{j=1}^n\}. \quad (6)$$

Assume that $\mathbf{a}_{i,\hat{S}}$ is the update of \mathbf{a}_i on the \hat{S} . Next, the truncated spectral method is exploited to make an initial estimate value \mathbf{z}^0 that is closer to the original sparse signal \mathbf{x} . Let \mathbf{Y} be a semidefinite Hermitian matrix, and the eigenvector $\tilde{\mathbf{z}}$ corresponding to the leading eigenvector of \mathbf{Y} is computed:

$$\mathbf{Y} = \frac{1}{m} \sum_{i=1}^m y_i \mathbf{a}_{i,\hat{S}} \mathbf{a}_{i,\hat{S}}^T \mathbf{I}_{\{|y_i| \leq \alpha_y^2 \lambda^2\}}, \quad (7)$$

where $\lambda^2 = \frac{1}{m} \sum_{i=1}^m y_i$, α_y is the predetermined truncation

threshold. The initial estimate can be obtained by $\mathbf{z}^0 = \lambda \tilde{\mathbf{z}}$.

Stage 2: reweighted gradient descent for sparse signal

For the t -th iteration of the reweighted gradient descent, the formula is given by:

$$\tilde{\mathbf{z}}^{t+1} = \mathbf{z}^t - \frac{\mu_t}{\lambda^2} \nabla f(\mathbf{z}^t), \quad (8)$$

where μ_t is variable step size.

By changing weights during the gradient descent, we can adaptively control the update direction. This not only accelerates the convergence rate but also contributes to obtaining a more optimal solution. The reweighted gradient is

$$\nabla f(\mathbf{z}^t) = \frac{1}{m} \sum_{i=1}^m w_i^t (|\mathbf{a}_i^T \mathbf{z}^t|^2 - y_i) (\mathbf{a}_i^T \mathbf{x}) \mathbf{a}_i^T, \quad (9)$$

and $\{w_i\}_{i=1}^m$ are determined by

$$w_i^t = \frac{1}{\left| |\mathbf{a}_i^T \mathbf{z}^t|^2 - |\mathbf{a}_i^T \mathbf{x}|^2 \right| + \eta}, \quad i = 1, \dots, m, \quad (10)$$

where \mathbf{z}^t is the optimal value in the t -th iteration, the proposed algorithm updates \mathbf{z}^t from an appropriate initialization \mathbf{z}^0 . η is a fixed parameter, and specific instructions regarding it are provided below. The weight $\{w_i\}_{i=1}^m$ vary with \mathbf{z}^t for each iteration. Therefore, a weight value is recalculated with each update.

To reduce the dimensional freedom and limit the searching domain, a thresholding operator \mathcal{H}_k is applied to \mathbf{z}^{t+1} .

$$\mathbf{z}^{t+1} = \mathcal{H}_k(\tilde{\mathbf{z}}^{t+1}). \quad (11)$$

This means that \mathbf{z}^{t+1} keeps the k elements of $\tilde{\mathbf{z}}^{t+1}$ with the largest absolute value, and the rest elements of $\tilde{\mathbf{z}}^{t+1}$ are zero. The algorithmic details of SRThWF can be summarized in Table I.

To intuitively establish the convergence of SRThWF, we define relative mean square error (RMSE) as follows:

$$\text{RMSE} := \frac{\text{dist}(\mathbf{z}, \mathbf{x})}{\|\mathbf{x}\|_2}, \quad (12)$$

where \mathbf{z} is the numerical estimation of \mathbf{x} , $\text{dist}(\mathbf{z}, \mathbf{x})$ represents the Euclidean distance from estimation value \mathbf{z} to the solution \mathbf{x} , $\text{dist}(\mathbf{z}, \mathbf{x}) := \min\{\|\mathbf{z} + \mathbf{x}\|_2, \|\mathbf{z} - \mathbf{x}\|_2\}$.

TABLE I
THE DETAILS OF THE PROPOSED ALGORITHM

Algorithm 1: SRThWF

1: Input:

$\{\mathbf{a}_i, y_i\}_{i=1}^m$: Gaussian random vectors and intensity values

k : the sparsity of \mathbf{x}

ε : the accuracy required

μ_t : the step size

α_y : truncation threshold

T : the maximum iterative time

η : the fixed parameter

2: initialization restricted on support recovery

\hat{S} includes indicators corresponding to the largest index of

$$k \text{ in } \sum_{i=1}^m \frac{y_i \mathbf{a}_i^2}{m}$$

$$\mathbf{Y} = \frac{1}{m} \sum_{i=1}^m y_i \mathbf{a}_i \mathbf{a}_i^T \mathbf{I}_{\{|y_i| \leq \alpha_y^2, \lambda^2\}}$$

$$\mathbf{z}^0 = \lambda \tilde{\mathbf{z}}$$

3: Reweighted gradient descent

For $\|\mathbf{z}^{t+1} - \mathbf{z}^t\| \geq \varepsilon$ and $t \leq T$ do

$$\mathbf{z}^{t+1} = \mathcal{H}_k \left\{ \mathbf{z}^t - \frac{\mu_t}{\lambda^2} \left(\frac{1}{m} \sum_{i=1}^m w_i^t (|\mathbf{a}_i^T \mathbf{z}^t|^2 - y_i)^2 (\mathbf{a}_i^T \mathbf{x}) \mathbf{a}_i^T \right) \right\},$$

$$\text{where } w_i^t = \frac{1}{\left| |\mathbf{a}_i^T \mathbf{z}^t|^2 - |\mathbf{a}_i^T \mathbf{x}|^2 \right| + \eta}, \quad i = 1, \dots, m,$$

$$t = t + 1$$

4: Output: $\tilde{\mathbf{z}}^t$: the final estimated signal

As depicted in Fig.1, different η have different convergence rates for SRThWF. Next, we can theoretically establish the convergence of the proposed algorithm.

B. Convergence Proof

Lemma 1. Let \mathbf{z}^t be the t -iteration value of the algorithm 1, S^{t+1} and \hat{S} are the support of \mathbf{z}^{t+1} and k -sparse real solution \mathbf{x} by SRThWF, respectively. Assume $\mathcal{L}^{t+1} = S^{t+1} \cup \hat{S}$, for all t , then $|\mathcal{L}^{t+1}| \leq 2k$.

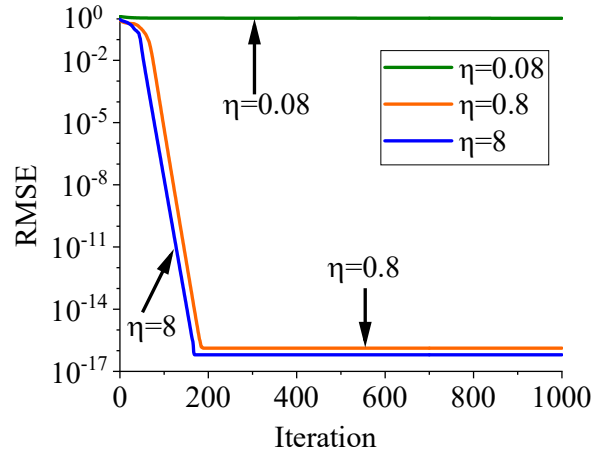


Fig.1. RMSE versus iteration for $\mathbf{x} \in R^{1000}$ with $m/n=3$ fixed

Lemma 2. Under the condition of Lemma 1, let $\mathbf{h}^t = \mathbf{z}^t - \mathbf{x}e^{j\theta^t}$, $\phi^t = \arg \min_{\phi \in [0, \pi]} \|\mathbf{z}^t - \mathbf{x}e^{j\phi}\|$. If $\mathcal{L}^{t+1} \cap (\mathcal{L}^t \setminus \mathcal{L}^{t+1}) = \emptyset$, then we have $\|\mathbf{h}_{\mathcal{L}^{t+1}}^t\| + \|\mathbf{h}_{\mathcal{L}^t \setminus \mathcal{L}^{t+1}}^t\| \leq \sqrt{2} \|\mathbf{h}^t\|$.

Theorem 3. Consider a k -sparse vector $\mathbf{x} \in \mathbb{R}^n$. $y_i = |\langle \mathbf{a}_i, \mathbf{x} \rangle|^2$, $i = 1, \dots, m$, $\mathbf{a}_i \in \mathbb{R}^n$ and \mathbf{a}_i are i.i.d. $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. If there exists a constant $0 < \nu < 1$ in t -iteration of the algorithm 1, we have

$$\|\mathbf{h}^{t+1}\| \leq \nu \|\mathbf{h}^t\|.$$

Proof. For the t -iteration of Algorithm 1, we have

$$\begin{aligned} \|\mathbf{h}^{t+1}\| &= \|\mathbf{z}^{t+1} - \mathbf{x}e^{j\theta^{t+1}}\| \leq \|\mathbf{z}_{\mathcal{L}^{t+1}}^{t+1} - \mathbf{x}_{\mathcal{L}^{t+1}}e^{j\theta^{t+1}}\| \\ &\leq \|\mathbf{z}_{\mathcal{L}^{t+1}}^{t+1} - \tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1}\| + \|\mathbf{x}_{\mathcal{L}^{t+1}}e^{j\theta^{t+1}} - \tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1}\|, \end{aligned} \quad (13)$$

where $\tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1} = \mathbf{z}_{\mathcal{L}^{t+1}}^{t+1} - \frac{\mu_t}{\lambda^2} \nabla f(\mathbf{z}^t)_{\mathcal{L}^{t+1}}$.

From the result of Lemma 1, $\|\mathbf{z}_{\mathcal{L}^{t+1}}^{t+1} - \tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1}\| \leq \|\mathbf{x}_{\mathcal{L}^{t+1}}e^{j\theta^{t+1}} - \tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1}\|$,

this substitute $\tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1} = \mathbf{z}_{\mathcal{L}^{t+1}}^{t+1} - \frac{\mu_t}{\lambda^2} \nabla f(\mathbf{z}^t)_{\mathcal{L}^{t+1}}$, then we have

$$\|\mathbf{h}^{t+1}\| \leq 2 \left\| \mathbf{x}_{\mathcal{L}^{t+1}}e^{j\theta^t} - \tilde{\mathbf{z}}_{\mathcal{L}^{t+1}}^{t+1} - \frac{\mu_t}{\lambda^2} \nabla f(\mathbf{z}^t)_{\mathcal{L}^{t+1}} \right\| = 2 \left\| \mathbf{h}_{\mathcal{L}^{t+1}}^t - \frac{\mu_t}{\lambda^2} \nabla f(\mathbf{z}^t)_{\mathcal{L}^{t+1}} \right\|.$$

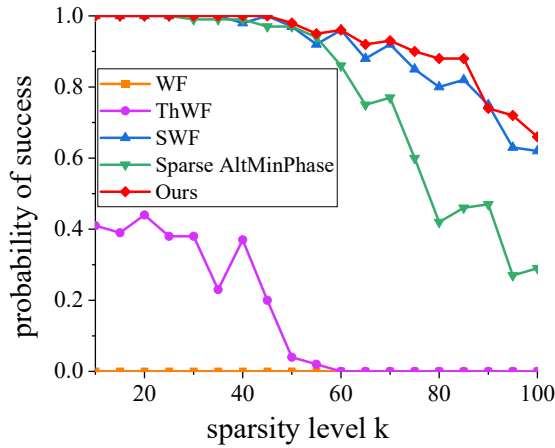
According algorithm 1, $w_i^t = \frac{1}{\left| |\mathbf{a}_i^T \mathbf{z}^t|^2 - |\mathbf{a}_i^T \mathbf{x}|^2 \right| + \eta}$, $i = 1, \dots, m$ and

$$\nabla f(\mathbf{z}^t)_{\mathcal{L}^{t+1}} = \frac{2}{m} \sum_{i=1}^m w_i^t (2(\mathbf{a}_i^T \mathbf{x})^2 \mathbf{a}_i^T \mathbf{h}^t + 3\mathbf{a}_i^T \mathbf{x}e^{j\theta^t} (\mathbf{a}_i^T \mathbf{h}^t)^2 + (\mathbf{a}_i^T \mathbf{h}^t)^3) \mathbf{a}_{i, \mathcal{L}^{t+1}},$$

we have

$$\begin{aligned} \|\mathbf{h}^{t+1}\| &\leq 2 \left\| \mathbf{h}_{\mathcal{L}^{t+1}}^t - \frac{\mu_t}{\lambda^2} \frac{2}{m} \sum_{i=1}^m w_i^t (2(\mathbf{a}_i^T \mathbf{x})^2 \mathbf{a}_i^T \mathbf{h}^t + 3\mathbf{a}_i^T \mathbf{x}e^{j\theta^t} (\mathbf{a}_i^T \mathbf{h}^t)^2 + (\mathbf{a}_i^T \mathbf{h}^t)^3) \mathbf{a}_{i, \mathcal{L}^{t+1}} \right\| \\ &\leq 2 \left\| \mathbf{h}_{\mathcal{L}^{t+1}}^t - \frac{4\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t (\mathbf{a}_i^T \mathbf{x})^2 \mathbf{a}_i^T \mathbf{h}_{\mathcal{L}^{t+1}}^t + \mathbf{h}_{\mathcal{L}^{t+1}}^t \right\| \\ &\quad + 12 \left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t \mathbf{a}_i^T \mathbf{x} (\mathbf{a}_i^T \mathbf{h}^t)^2 \mathbf{a}_{i, \mathcal{L}^{t+1}} \right\| + 4 \left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t (\mathbf{a}_i^T \mathbf{h}^t)^3 \mathbf{a}_{i, \mathcal{L}^{t+1}} \right\| \\ &\quad + 8 \left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t (\mathbf{a}_i^T \mathbf{x})^2 \mathbf{a}_i^T \mathbf{h}_{\mathcal{L}^t \setminus \mathcal{L}^{t+1}}^t \mathbf{h}_{\mathcal{L}^t \setminus \mathcal{L}^{t+1}}^t \mathbf{a}_{i, \mathcal{L}^{t+1}} \right\|. \end{aligned} \quad (14)$$

If there exists a constant w_i , $\beta_2 < w_i < \beta_1$, $i = 1, \dots, m$, and from [16] and [17], there exist $0 < \delta_1 < 1$, we get the following inequality


 Fig.2. Probability of success versus sparsity level k

$$\begin{aligned} & \left\| \mathbf{h}'_{\mathcal{L}^{t+1}} - \frac{4\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t (\mathbf{a}_i^T \mathbf{x})^2 \mathbf{a}_{i,\mathcal{L}^{t+1}}^T \mathbf{h}'_{\mathcal{L}^{t+1}} \mathbf{a}_{i,\mathcal{L}^{t+1}} \right\| \quad (15) \\ & \leq \max \left\{ 1 - \frac{4\mu_t}{\lambda^2} \beta_2 (3 - \delta_1) \|\mathbf{x}\|^2, \frac{4\mu_t}{\lambda^2} \beta_1 (3 + \delta_1) \|\mathbf{x}\|^2 - 1 \right\} \|\mathbf{h}'_{\mathcal{L}^{t+1}}\|. \end{aligned}$$

Then, there exists $0 < \delta_2 < 1$ and $w_i < \beta_1, i = 1, \dots, m$, we obtain

$$\left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m (\mathbf{a}_i^T \mathbf{x})^2 w_i^t \mathbf{a}_{i,\mathcal{L}^{t+1}}^T \mathbf{h}'_{\mathcal{L}^{t+1}} \mathbf{a}_{i,\mathcal{L}^{t+1}} \right\| \leq \frac{\mu_t}{\lambda^2} \beta_1 \delta_2 \|\mathbf{x}\|^2 \|\mathbf{h}'_{\mathcal{L}^{t+1}}\| \quad (16)$$

Similarly, for any fixed $0 < \delta_3 < 1$ and $w_i < \beta_1, i = 1, \dots, m$, we have (17) and (18)

$$\left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t \mathbf{a}_i^T \mathbf{x} (\mathbf{a}_i^T \mathbf{h}^t)^2 \mathbf{a}_{i,\mathcal{L}^{t+1}} \right\|^2 \leq \frac{\sqrt{17}\mu_t \delta_3}{\lambda^2} \beta_1 \|\mathbf{x}\|^2 \|\mathbf{h}^t\|, \quad (17)$$

$$\left\| \frac{\mu_t}{m\lambda^2} \sum_{i=1}^m w_i^t \mathbf{a}_i^T \mathbf{x} (\mathbf{a}_i^T \mathbf{h}^t)^2 \mathbf{a}_{i,\mathcal{L}^{t+1}} \right\|^2 \leq \frac{\sqrt{17}\mu_t \delta_3}{\lambda^2} \beta_1 \|\mathbf{x}\|^2 \|\mathbf{h}^t\|. \quad (18)$$

Combining from (15) to (18), we obtain

$$\begin{aligned} \|\mathbf{h}^{t+1}\| & \leq 2 \max \left\{ 1 - \frac{2\mu_t}{\lambda^2} \beta_2 (3 - \delta_1) \|\mathbf{x}\|^2, \frac{2\mu_t}{\lambda^2} \beta_1 (3 + \delta_1) \|\mathbf{x}\|^2 - 1 \right\} \|\mathbf{h}'_{\mathcal{L}^{t+1}}\| \\ & \quad + \frac{8\mu_t \delta_2 \beta_1}{\lambda^2} \|\mathbf{x}\|^2 \|\mathbf{h}'_{\mathcal{L}^{t+1}}\| + \frac{12\sqrt{17}\mu_t \delta_3 \beta_1}{\lambda^2} \|\mathbf{x}\|^2 \|\mathbf{h}^t\| \\ & \quad + \frac{4\sqrt{17}\mu_t \delta_3 \beta_1}{\lambda^2} \|\mathbf{x}\|^2 \|\mathbf{h}^t\|. \end{aligned} \quad (19)$$

From **Lemma 2**, $\|\mathbf{h}'_{\mathcal{L}^{t+1}}\| \leq \sqrt{2} \|\mathbf{h}^t\|$ holds, it is substituted to (18), and following the reference [13] there exist $0 < \delta_4 < 1$, $(1 - \delta_4)\lambda^2 \leq \|\mathbf{x}\|^2 \leq (1 + \delta_4)\lambda^2$ holds. We have the following conclusion,

$$\begin{aligned} \|\mathbf{h}^{t+1}\| & \leq \left[\sqrt{2} \max \left\{ 2 \max \left\{ 1 - 4\mu_t \beta_2 (1 - \delta_4) (3 - \delta_1), \right. \right. \right. \\ & \quad \left. \left. 4\mu_t \beta_1 (1 + \delta_4) (3 + \delta_1) - 1, 8\mu_t \delta_2 \beta_1 (1 + \delta_4) \right\} \right. \\ & \quad \left. + (12\sqrt{17} + 4\sqrt{17}\delta_3) (1 + \delta_4) \mu_t \delta_3 \beta_1 \right] \|\mathbf{h}^t\|. \end{aligned} \quad (20)$$

By choosing a feasible μ_t , there exists $v (0 < v < 1)$, then

$$\|\mathbf{h}^{t+1}\| \leq v \|\mathbf{h}^t\|. \quad (21)$$

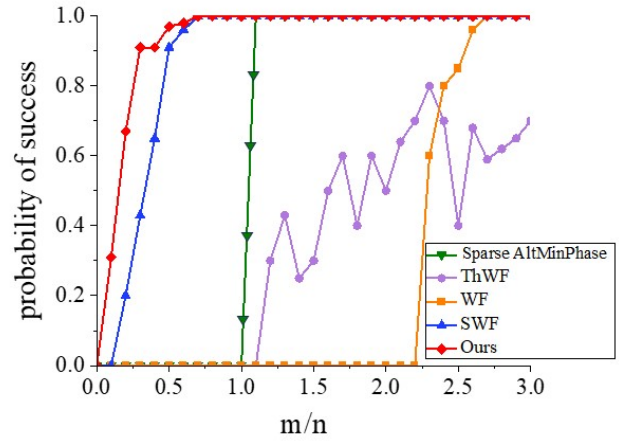
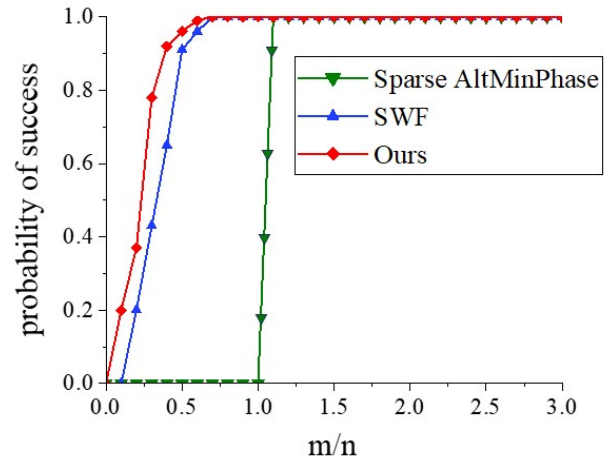
This completes the proof of the proposed algorithm. \square

Specifically, if $\{w_i\}_{i=1}^m = 1$, this is the algorithm in [14].

III. EXPERIMENT AND ANALYSIS

A. Evaluation Metrics and Experimental Settings

To assess the practical effectiveness of the SRThWF algorithm, we conduct several numerical experiments. All


 Fig.3. Probability of success versus sample rate m/n

 Fig.4. Probability of success versus sample rate m/n

tests are carried out on a system running the win10x64-bit operating system, with an AMD Ryzen 5 pro 2.0 GHZ CPU, and 8GB RAM, the experimental platform is MATLAB 2018b.

We employ various metrics to assess the performance of the algorithm, such as the probability of success over 100 independent Monte Carlo trials [18], running time, and the RMSE. A trial is considered successful if the RMSE is below 10^{-5} .

In all simulation tests, we set $n = 1000$, truncation threshold $\alpha_y = 3$, $\mathbf{x} \in \mathbb{R}^n$, $\{\mathbf{a}_i\}_{i=1}^m$ are i.i.d. $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and run 100 power iterations to obtain the initialization. Similar to reference [6], all simulation tests opt for a variable step size $\mu_t = \min((1 - e^{-t/330}), 0.22)$ to prevent the algorithm from getting trapped in a local minimum during its initial stages.

B. Experiment Analysis

Let the sample rate $m/n = 1.5$, and the sparsity level k range from 10 to 100. As shown in Fig.2, it is evident that WF consistently fails to reconstruct the original signal. When $k < 60$, ThWF can recover the signal but the probability of success is less than 50%. On the other hand, SWF, sparse AltMinPhase, and the proposed algorithm can successfully recover the original signal by approximately 90% when sparsity level $k < 55$. With the sparsity level k increasing, especially $k > 60$, SRThWF algorithm outperforms SWF and sparse AltMinPhase with a better probability of success.

In Fig.3, $k = 10$ is known, and the sample rate takes value $\{1, 2, 3\}$. It can be seen that ThWF can recover the signal even when $m > 1.1n$ in certain cases, but it cannot stably achieve a 100% probability of success for all sample rates. Sparse AltMinPhase achieves a 100% probability of success only when $m \geq 1.1n$. WF needs $m \geq 2.6n$ to achieve a 100% probability of success. Both SRThWF and SWF achieve a 100% probability of success when $m \geq 0.7n$, but SRThWF consistently demonstrates a higher probability of success at lower sample rates.

Suppose the sparsity level k is unknown. Set k is the theoretically tolerable upper limit $\lceil \sqrt{n} \rceil$, and keep other experimental settings the same as those for the known sparsity condition. From Fig.4, we can observe that SRThWF still has the same high probability of success when k is unknown correctly. This observation highlights the benefits of integrating weighted factors into sparse phase retrieval problem.

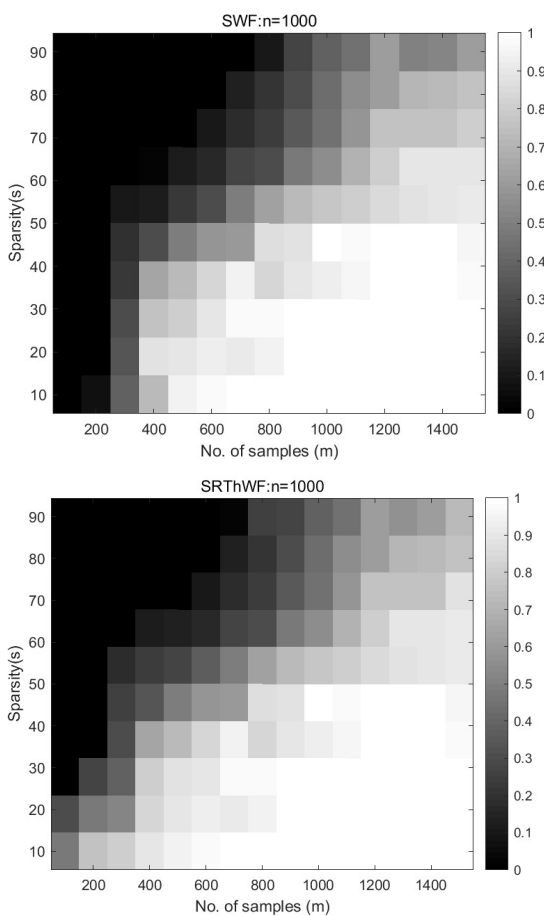


Fig.5. Phase transition for different algorithms.

Furthermore, Fig.5 depicts the phase transitions of both the advanced SWF algorithm and the SRThWF algorithm. The sparsity levels k ranges from 10 to 100, and the sample size m ranges from 100 to 1500, with grid size of 100. The probability of success is depicted by the gray level of each block: a white block signifies a 100% successful recovery rate, a black block represents 0%, and a gray block represents a rate between 0% and 100%. The results indicate that the proposed algorithm demands fewer measurements than the SWF algorithm while achieving the same successful recovery rate. In light of this, it can be confidently stated that the

SRThWF algorithm surpasses the state-of-the-art SWF algorithm.

In Table II, the bold font represents the current best value for different sample ratios. Iteration count and running time are compared between the proposed SRThWF algorithm and SWF when the RMSE is below 10^{-15} . SRThWF takes 50 fewer iterations than SWF. Not only that, SRThWF is 0.8 seconds faster than SWF in terms of running time. Notably, weights are added to the gradient descent to adaptively control the updated direction of parameters. Hence, the SRThWF algorithm outperforms the SWF algorithm in convergence rate and running time.

TABLE II
THE COMPUTATIONAL COST BETWEEN SWF AND SRThWF FOR THE DIFFERENT SAMPLE RATE.

m/n	Algorithm	iterations	Time(s)
1	SWF	140	1.419
	Ours	86	0.82956
2	SWF	113	2.167
	Ours	65	1.1826
3	SWF	98	2.698
	Ours	49	1.5453

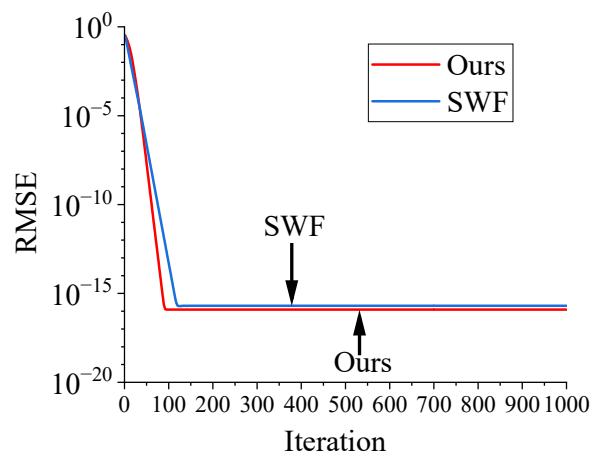


Fig.6. Comparison of convergence rate for different algorithms

At a sample rate $m/n = 3$ and with a sparsity level $k = 10$, as shown in Fig.6, it becomes evident that the SRThWF algorithm outperforms SWF with respect to convergence rate. Specifically, the proposed algorithm achieves an RMSE of 10^{-15} in only 76 iterations, in stark contrast to SWF, which demands 128 iterations to attain the same RMSE level.

For $k = 10$, to test the robustness of SRThWF in the presence of the Gaussian random noise, the RMSE varies from the different signal-to-noise (SNR) values in dB. The SNR in dB ranges from 5 dB to 55 dB, and the ratio m/n takes values 1 to 3. Fig.7 shows the robustness of SRThWF under Gaussian random noise for different sample rates.

Table III lists the RMSE values obtained at different sample rates for various SNR values. Especially, the RMSE of the SRThWF algorithm is below 7.35×10^{-5} when the SNR value is higher than 45 dB and $m/n = 1$. As the sample rate increases to 2, the SNR value increases to 50, and the RMSE of the proposed algorithms is 7.22×10^{-5} . When the

SNR is 45 dB, and $m/n = 3$, the RMSE of the proposed algorithms is 4.40×10^{-5} . From Table III, the numerical results analysis also confirms the robustness of the proposed algorithm.

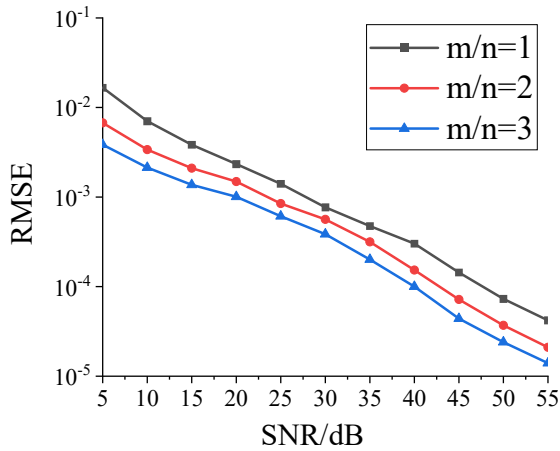


Fig.7. RMSE versus SNR/dB for SRThWF with the AWGN model.

TABLE III
SRThWF WITH DIFFERENT SAMPLE RATES AND SNR ON RMSE.

m/n	1	2	3
5	1.67×10^{-2}	6.70×10^{-3}	3.85×10^{-3}
10	7.04×10^{-3}	3.39×10^{-3}	2.14×10^{-3}
15	3.84×10^{-3}	2.10×10^{-3}	1.37×10^{-3}
20	2.34×10^{-3}	1.49×10^{-3}	1.01×10^{-3}
25	1.14×10^{-3}	8.47×10^{-4}	6.11×10^{-4}
30	7.68×10^{-4}	5.63×10^{-4}	3.85×10^{-4}
35	4.75×10^{-4}	3.16×10^{-4}	2.01×10^{-4}
40	3.02×10^{-4}	1.53×10^{-4}	1×10^{-4}
45	1.44×10^{-4}	7.22×10^{-5}	4.40×10^{-5}
50	7.35×10^{-5}	3.7×10^{-5}	2.40×10^{-5}
55	4.20×10^{-5}	2.10×10^{-5}	1.40×10^{-5}

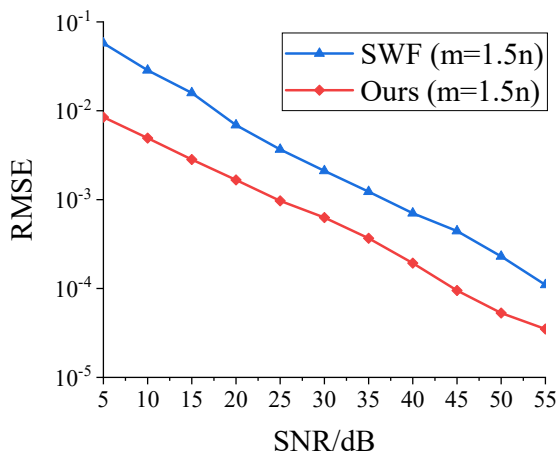


Fig.8. RMSE versus SNR/dB for different algorithms

Finally, we assess the robustness of different algorithms by varying the SNR from 5dB to 55dB while keeping $k = 10$ and $m/n = 1.5$ constant. The results are illustrated in Fig.8. It is evident that the SRThWF algorithm exhibits greater resilience to noise compared to SWF. Moreover, the proposed algorithm consistently achieves a lower RMSE than the SWF algorithm.

IV. CONCLUSION

This paper presents an adaptive reweighted thresholded algorithm designed to tackle the sparse PR problem. The proposed algorithm adopts the truncated spectral initialization strategy to generate a reasonable initial estimate. During the gradient descent process, the application of weights can assist the proposed algorithm in searching for the global minimum solution and accelerating the convergence rate. Experimental results show that compared with other state-of-the-art algorithms, the proposed algorithm has a higher probability of success even when the sparsity level is unknown, a faster convergence rate, and less running time. Moreover, it maintains robustness in the presence of Gaussian noise. Going forward, we will further investigate Fourier ptychographic microscopy for the sparse PR problem.

REFERENCES

- [1] X. Tao, J. Zhang, P. Sun, C. Wang, C. N. Tao, R. M. Wu, Z. R. Zheng, "Phase-coded speckle illumination for laser Fourier ptychographic microscopy," *Optics Communications*, vol. 498, pp. 127199, 2021.
- [2] G. Zhang, J. Li, K. Deng, "Reweighted L1-norm regularized phase retrieval for x-ray differential phase contrast radiograph," *The Review of Scientific Instruments*, vol. 93, no. 4, pp. 043706, 2022.
- [3] Y. Guo, Y. Wu, Y. Li, X. Rao, C. Rao, "Deep phase retrieval for astronomical Shack-Hartmann wavefront sensors," *Monthly Notices of the Royal Astronomical Society*, vol. 510, no. 3, pp. 4347-4354, 2022.
- [4] G. Sinclair, J. Leach, P. Jordan, "Interactive application in holographic optical tweezers of a multi-plane Gerchberg-Saxton algorithm for three-dimensional light shaping," *Optics Express*, vol. 12, no. 8, pp. 1665-1670, 2004.
- [5] L. Li, C. Cheng, D. G. Han, Q.Y. Sun, G. M. Shi, "Phase retrieval from multiple-window short-time Fourier measurements," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 372-376, 2017.
- [6] E. J. Candes, X. Li, M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985-2007, 2015.
- [7] Y. Chen, E. J. Candès, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," *Communications on Pure and Applied Mathematics*, vol. 70, no. 5, pp. 822-883, 2017.
- [8] Z. Yuan, H. Wang, "Phase retrieval via reweighted Wirtinger flow," *Applied Optics*, vol. 56, no. 9, pp. 2418-2427, 2017.
- [9] Y. Zhang, Y. Liu, X. Zhang, "A variable stepsize sparsity adaptive matching pursuit algorithm," *IAENG International Journal of Computer Science*, vol. 48, no. 3, pp. 770-775, 2021.
- [10] L. Li, X. L. Qiu, M. L. Jing, S. S. Pu, "Block compressed sensing image reconstruction via untrained network priors," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 505-511, 2023.
- [11] H. Ohlsson, Y. C. Eldar, "On conditions for uniqueness in sparse phase retrieval," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1841-1845, 2014.
- [12] P. Netrapalli, P. Jain, S. Sanghavi, "Phase retrieval using alternating minimization," *IEEE Transactions on Signal Processing: A Publication of the IEEE Signal Processing Society*, vol. 63, no.18, pp. 4814-4826, 2015.
- [13] T. T. Cai, X. Li, Z. Ma, "Optimal rates of convergence for noisy sparse phase retrieval via thresholded Wirtinger flow," *The Annals of Statistics*, vol. 44, no. 5, pp. 2221-2251, 2016.
- [14] Z. Yuan, H. Wang, Q. Wang, "Phase retrieval via sparse Wirtinger flow," *Journal of Computational and Applied Mathematics*, vol 355, pp. 162-173, 2019.

- [15] G. Wang, L. Zhang, G. B. Giannakis, "Sparse phase retrieval via truncated amplitude flow," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 479-491, 2017.
- [16] J. Sun, Q. Qu, J. Wright, "A geometric analysis of phase retrieval," *Foundations of Computational Mathematics*, vol. 18, no. 5, pp. 1131-1198, 2018.
- [17] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *Compressed Sensing*, pp. 210-268, 2012.
- [18] D. P. Wipf, B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3704-3716, 2007.